

PATENT APPLICATION
Docket No.: 6647-030
Client Ref. No.: IDR-533

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Boyd H. TIMOTHY, et al.

Serial No.: 10/066,465 Examiner: Manglesh M. PATEL

Filed: January 30, 2002 Art Unit: 2178

Confirmation No.: 3498

For: METHOD AND APPARATUS TO DYNAMICALLY PROVIDE WEB
CONTENT RESOURCES IN A PORTAL

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF JAMES MARK NORMAN

I, JAMES MARK NORMAN, hereby declare:

1. I have been employed as an engineer at Novell, Inc. in the Portal Services Group since prior to 2001. While working in the Portal Services Group in 2001, I participated in the conception and implementation of Novell Portal Services 1.0, Service Pack 1 ("NPS Service Pack 1"). I am a co-inventor of the subject matter described in related U.S. Patent Application Serial No. 10/066,368 as well as this U.S. Patent Application Serial No. 10/066,465. I was omitted from being named as an inventor in this patent application by an inadvertent error, which is in the process of being rectified.

2. Development of NPS Service Pack 1 began before May 18, 2001. The document named Portal I18N Architecture.doc and titled "Novell Portal Service I18N Architecture" describes features implemented in NPS Service Pack 1 and relevant to this patent application.

The document named Portal I18N Architecture.doc was first saved into the document management system used by Novell, Inc. before May 18, 2001. A true copy of the document named Portal I18N Architecture.doc, as it existed before May 18, 2001, is attached hereto as Exhibit A. The document management system used by Novell, Inc. is typical of document management systems and source code systems used in the industry, and its use by software developers at Novell, Inc. is a standard business practice at Novell, Inc. A screenshot, showing that the document named Portal I18N Architecture.doc was checked in to the document management system used by Novell, Inc. before May 18, 2001, is attached hereto as Exhibit B.

3. Implementation of the features described in the document named Portal I18N Architecture.doc proceeded at a normal pace. A first implementation of software containing the features described in the document named Portal I18N Architecture.doc was completed by June 5, 2001. Exhibit C is a screenshot of the folder `$/NPS1.5/nps/WEB-INF/src/com/novell/nps/i18n`, showing files saved in this folder; the files in this folder implement the features described in the document named Portal I18N Architecture.doc. Exhibit C also shows details of `SystemResourceFileMap.java`, one of the files in this folder and implementing the features of the document named Portal I18N Architecture.doc, and shows that this file was created on May 29, 2001.

4. NPS Service Pack 1 was completed in July 2001 and was commercially released on July 24, 2001. A true copy of the news brief announcing the release of NPS Service Pack 1 is attached hereto as Exhibit D. All of the features of the claims of this U.S. Patent Application Serial No. 10/066,465 were reduced to practice in NPS Service Pack 1 as released and available for download on July 24, 2001.

5. The claims of U.S. Patent Application Serial No. 10/066,465 are rejected under 35 U.S.C. § 103(a) over U.S. Patent Application Publication No. 2002/0174150 to Dang et al. (“Dang”) in view of non-patent publication “CSS Mobile Profile 1.0” to Wugofski et al. (“Wugofski”). Dang has an effective filing date in the United States of May 18, 2001. Wugofski has a publication date of October 24, 2001.

6. The claims of U.S. Patent Application Serial No. 10/066,465 are supported by the document named Portal 118N Architecture.doc (Exhibit A). For example, the support for claims 1, 16, 31, 39, and 47 are described below (the remaining claims are similarly supported):

Claim 1. An apparatus for presenting content to a user, comprising:
a plurality of layout strings files;
a plurality of layout information files to describe how a layout string is displayed for a unique combination of a language and a device; and
a computer to store the layout strings files and the layout information files.

Claim 16. A computer-implemented method for displaying content to a user, comprising:
locating a layout information file from a plurality of layout information files specifying how a layout string is to be presented to the user for a unique combination of a language and a device;
locating one of a plurality of layout strings files storing the layout string; and
presenting the layout string to the user according to the located layout information file.

Claim 31. One or more computer-readable media containing a program to display content to a user, comprising:
location software to locate a layout information file from a plurality of layout information files specifying how a layout string is to be presented to the user for a unique combination of a language and a device;
location software to locate one of a plurality of layout strings files storing the layout string; and
presentation software to present the layout string to the user according to the located layout information file.

Claim 39. An article comprising:
a computer-readable modulated carrier signal;

means embedded in the signal for locating a layout information file from a plurality of layout information files specifying how a layout string is to be presented to a user for a unique combination of a language and a device;

means embedded in the signal for locating one of a plurality of layout strings files storing the layout string; and

means embedded in the signal for presenting the layout string to the user according to the located layout information file.

Claim 47. A computer-implemented method for using a selected context to display content to a user, comprising:

locating a layout information file from a plurality of layout information files specifying how the content is to be presented to the user for a unique combination of a language and a device;

locating a layout strings file storing a layout string in the selected context; and

presenting the content and the layout string in the selected context to the user according to the located layout information file.

Each of the elements of the foregoing claims is supported in the specification of the present patent application as follows:

Referring to FIG. 8 of the original Patent Application drawings, a file hierarchy structure is shown including a plurality of layout strings files and a plurality of layout information files. Page 12, lines 26-30 of the Patent Application describes the layout strings files of FIG. 8 as follows:

Main_lang.xml 855 and its siblings store the layout strings (described above with reference to FIGs. 4-5). That is, main_lang.xml 855 stores the default layout strings, whereas main_lang_en_US.xml (860), main_lang_en_UK.xml (865), and main_lang_es_ES.xml (870) store the American English, British English, and Spanish layout strings, respectively. A person skilled in the art will recognize there can be other layout string files as well.

Page 11, lines 25-28 of the patent application describes the layout information files (LIFs) of FIG. 8 as follows:

Skins 810 is the subdirectory storing information about each skin (i.e., LIF) available for the device. Each skin is assigned a name, one of which is Default 815. For example, in FIG. 8, Alternative Skin 1 875 is one alternative skin. Other alternative skins can also be present.

Page 6, lines 20-24 of the Patent Application describes how a layout information files can describe how a layout string is displayed for a particular language.

A person skilled in the art will recognize that there are times when LIFs can be language dependent. For example, some languages are displayed right-to-left (such as Japanese, Hebrew, or Arabic). For these languages, a LIF that has the information displayed in a different presentation can be preferable. How LIFs are associated with particular languages is discussed further with reference to FIG. 7A below.

Page 6, lines 5-10 of the Patent Application describes how a layout information file can describe how a layout string is displayed for a particular device.

FIG. 4 shows one of the gadgets of FIG. 2 with two different layouts, according to an embodiment of the invention. In FIG. 4, layout information file (LIF) 405 specifies one layout for gadget 210, and LIF 455 specifies an alternative layout for gadget 210. (A layout information file is sometimes called a *skin*.) For example, LIF 405 might represent the default layout for gadget 210, whereas LIF 455 might represent a layout of gadget 210 for portable devices (where space is at a premium).

Each of the elements of claims 1, 16, 31, 39, and 47 is also supported in Exhibit A as follows:

Figure 1, on page 4 of Exhibit A, shows a file hierarchy similar to the file hierarchy shown in FIG. 8 of the Patent Application, with a plurality of layout strings files and layout information files.

Pages 4-5 of Exhibit A describes Gadget Stylesheets and corresponds to the above-quoted description of layout information files from the patent application at page 11, lines 24-28 and page 6, lines 5-10 and 20-24:

In the general case, a gadget will write one layout stylesheet for each skin (look) that it provides in the following directory:

`<gadget>/skins/<skin name>/devices/<device>/`

In our example (Fig. 1), we have created the stylesheet:

`com.novell.nps.gadgets.GadgetX/skins/default/devices/default/main.xsl`

Additional localized layout stylesheets may be defined using the localized file naming described above. This is useful in situations in which the portal must support multiple languages where the actual layout of a page must be different for two or more languages or locales. For example, if a portal needed to support both Japanese and English, the designer may wish to create a different layout than for Japanese users because Japanese is read left to right. If the original layout stylesheet is named *main.xsl*, this could be accomplished by defining the following layout stylesheet for Japanese:

`<gadget>/skins/<skin name>/devices/<device>/main_jp_JP.xsl`

To support the English portion that this portal would need to provide, the designer could either define an additional stylesheet for English in the same directory or allow the portal to default to the original *main.xsl* stylesheet.
(emphasis in original).

Page 5 of Exhibit A describes Gadget String Files and corresponds to the above-quoted description of the layout strings file of the patent application at page 12, lines 26-30.

Novell Portal Services uses XSL Stylesheets to store the localized strings needed by gadget layout stylesheets. Inside each language stylesheet, a gadget must define globally unique XSL variables that will be referenced in the layout stylesheets.

Each gadget requires an XSL/Language XSL file pair to provide the correct language and locale information for each user who authenticates to the portal. A gadget should define a group of Language XSL files for each language it plans to support. These files should follow the same naming pattern described earlier in this document.

For a basic and simple implementation, these files should be created and stored in the gadget's *skins/default/devices/default/* directory. In our example gadget (**Fig. 1**), GadgetX has defined *main_lang_[language code]_[country code].xsl* in the directory:

`com.novell.nps.gadgets.GadgetX/skins/default/devices/default/`

As previously described for gadget stylesheets, the portal includes a mechanism of providing additional levels of granularity as needed.

One example of when this is needed is when the portal must support both large and small display devices. It may be desired to provide detailed descriptions in Spanish when a user is using Internet Explorer 5 on their desktop computer, but short explanations when they login using a PocketPC device. In this scenario, two files with the same name are required in the following directories:


`<gadget>/skins/<skin name>/devices/ie5/main_lang_es_ES.xsl`
`<gadget>/skins/<skin name>/devices/pocketpc/main_lang_es_ES.xsl`

(emphasis in original).

Thus, the document named Portal I18N Architecture.doc discloses a plurality of layout information files, a plurality of layout string files, and presenting a layout string according to a layout information file, and therefore supports claims 1, 16, 31, 39, and 47. The remaining claims are similarly supported by the document named Portal I18N Architecture.doc.

9. Because the document named Portal I18N Architecture.doc supports the claims in U.S. Patent Application Serial No. 10/066,465, the invention was conceived before May 18, 2001, the effective filing date of Dang, and diligently embodied in NPS Service Pack 1 as released on July 24, 2001.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.


James Mark Norman

Dated: March 21, 2007

Novell Portal Services I18N Architecture

Abstract

This document will describe the internationalization architecture provided by Novell Portal Services Version 1.1. The portal provides the ability to obtain the user's desired language and locale information from the directory. This allows dynamic support of users' language and locale in portal web pages. Using the language and locale information found in the directory, the portal's main stylesheets and each gadget's stylesheet will be dynamically selected.

Language Determination

The purpose of determining the language is to come up with a list of languages and locales for the current user. The portal will compile this list base upon the strategy specified by the administrator. Currently, this search strategy is configured in the Portal Configuration Object (PCO) and/or the PortalServlet.properties (PSP) file.

In addition the portal contains a Default Portal Language. This is also configured in the PCO and/or PSP. The default language will be enUS. The language information will be published as part of the portal's `<SessionInfo>` tag to enable any gadget's layout stylesheet to query the language being used in the current session.

These lists are based on the ISO 639 and 3166 standards for languages and country codes. More information on these standards can be found at:

<http://www.unicode.org/unicode/onlinedat/languages.html>
<http://www.unicode.org/unicode/onlinedat/countries.html>

Once the portal determines the languages for a given user's session, it is returned to Gadget Manager. Gadget Manager uses this information in compiling the main stylesheet for each user session. The structure and layout required by this process will be described later in this document.

Language Determination Search Strategies

The portal ships with three search strategies. It is possible to add more search strategies in the future when necessary. Each strategy returns a list of possible languages and locales using the ISO Standard two-character notation. The portal then combines the list and returns a prioritized list of languages. Duplicates entries will not be added to the list.

The current strategies are:

1. Browser language. This strategy returns a list of the currently supported languages specified by the browser. Most browsers already use the ISO Standard.
2. Traditional NDS Language. This returns a list of the languages that are currently specified in the user's NDS Object. These languages are translated into ISO Standard before being returned.
3. Portal Language. This is a new setting on the portal user's object. This information is stored in the bhConfig setting. This is a list of languages ordered by priority. The portal's Administration and Configuration pieces have been changed to support this.

A default search strategy setting specifies the order of strategies to be used. The default search strategy will be set to:

2, 3

This will inform the portal to look in NDS first for any language information and then use the Portal's Default Language.

Other examples are listed here to illustrate how this setting can be used.

Example 1: 1, 2, 3

Using this combination of search strategies first looks for languages specified by the browser, then the traditional NDS language attribute, and finally the Default Portal Language.

If the browser's language is *deDE*, NDS contains *NIHONGO*, the Portal Language is *enUS*, and the default portal language is also *enUS*, the returned list would be: *deDE*, *jp*, *enUS*. Notice that the duplicate *enUS* is not added to the list.

Example 2: 3

This example only looks for the Portal Language and then also attaches the Default Portal Language.

If the Portal Language is set to *deDE* and the Default Portal Language is *enUS*, the returned list would be: *deDE*, *enUS*.

Gadget Directory Overview

Fig. 1 is a typical gadget's directory structure in Novell Portal Services. We recommend this same structure for every gadget that will be built for use in the portal. This structure enables the following:

1. Localization of the strings used in the gadget's layout stylesheet

2. Localization of the layout stylesheets themselves
3. Ability for a gadget to support different skins

One of the goals in the localization of each gadget's strings is to eliminate file duplication. That is to say that many different layout stylesheets should be able to utilize the same string file.

It is possible, however, for each gadget to define multiple string files if necessary. This will be useful, for example, when a gadget desires to provide a lengthy description for large screen devices and brief descriptions for small screen devices like a PDA.

Note: The portal's system stylesheets directory will follow the same directory structure outlined in this document for managing the different skins and localized files.

Localized File Naming

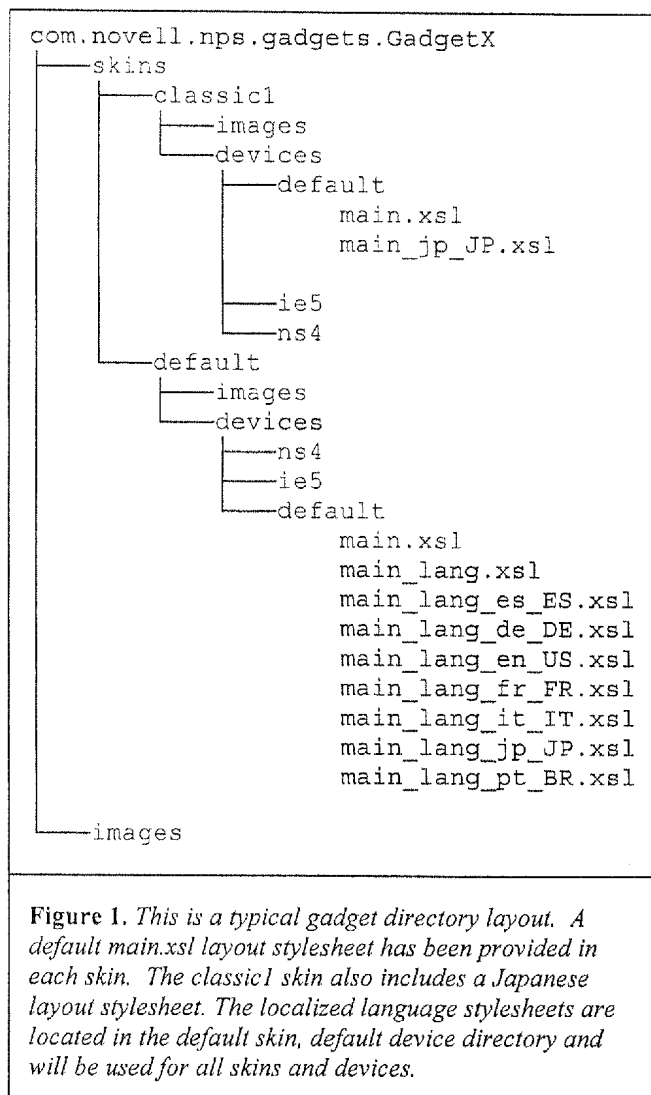
Throughout this document, we will refer to the localized naming of files. Novell Portal Services adheres to the ISO 639 Standard for language codes and ISO 3166 Standard for country codes for naming localized files. The portal will automatically use localized layout and language stylesheet files when available. These files will be named using the following pattern:

[non-localized filename]_[language code]_[country code].[extension]

For example, the following filename would be created when localizing *main.xsl* for the English language in the United States:

main_en_US.xsl

The same pattern is used for all localized files for different languages and countries.



Gadget Stylesheets

In the general case, a gadget will write one layout stylesheet for each skin (look) that it provides in the following directory:

```
<gadget>/skins/<skin name>/devices/<device>/
```

In our example (**Fig. 1**), we have created the stylesheet:

```
com.novell.nps.gadgets.GadgetX/skins/default/devices/default/main.xml
```

Additional localized layout stylesheets may be defined using the localized file naming described above. This is useful in situations in which the portal must support multiple languages where the actual layout of a page must be different for two or more languages or locales. For example, if a portal needed to support both Japanese and English, the designer may wish to create a different layout than for Japanese users because Japanese is

read left to right. If the original layout stylesheet is named *main.xml*, this could be accomplished by defining the following layout stylesheet for Japanese:

```
<gadget>/skins/<skin name>/devices/<device>/main_jp_JP.xml
```

To support the English portion that this portal would need to provide, the designer could either define an additional stylesheet for English in the same directory or allow the portal to default to the original *main.xml* stylesheet.

Gadget String Files

Novell Portal Services uses XSL Stylesheets to store the localized strings needed by gadget layout stylesheets. Inside each language stylesheet, a gadget must define globally unique XSL variables that will be referenced in the layout stylesheets.

Each gadget requires an XSL/Language XSL file pair to provide the correct language and locale information for each user who authenticates to the portal. A gadget should define a group of Language XSL files for each language it plans to support. These files should follow the same naming pattern described earlier in this document.

For a basic and simple implementation, these files should be created and stored in the gadget's *skins/default/devices/default/* directory. In our example gadget (**Fig. 1**), GadgetX has defined *main_lang_[language code]_[country code].xml* in the directory:

```
com.novell.nps.gadgets.GadgetX/skins/default/devices/default/
```

As previously described for gadget stylesheets, the portal includes a mechanism of providing additional levels of granularity as needed.

One example of when this is needed is when the portal must support both large and small display devices. It may be desired to provide detailed descriptions in Spanish when a user is using Internet Explorer 5 on their desktop computer, but short explanations when they login using a PocketPC device. In this scenario, two files with the same name are required in the following directories:

```
<gadget>/skins/<skin name>/devices/ie5/main_lang_es_ES.xml  
<gadget>/skins/<skin name>/devices/pocketpc/main_lang_es_ES.xml
```

The default device and default skin

The *devices/default* and *skins/default* directories are default implementations. When the portal fails to find localized stylesheets or string files in a given device/skin directory, it will default to use the ones found in the *devices/default* or *skins/default* directories. If the portal cannot find the needed file in these default directories, the process will fail and needs to be fixed by the administrator. This process will be described in more depth later.

Managing Image Files

Each gadget contains an *images* directory that can be used to store images specific to the gadget. In addition, each skin within a gadget also has an *images* directory. This should be used for graphics specific to a particular skin. It is the responsibility of the person writing the gadget layout stylesheet to reference the images in their proper location.

Gadgets may use localized image files. By this, we mean that an image can contain a graphic which is either specific to a specific language, locale or both. In this case, the gadget writer is responsible for creating and referencing the different graphic files for the different languages and locales. To maintain organization, a gadget writer can adopt the same ISO standards for languages and country codes to name graphics.

To avoid creating multiple layout stylesheets, a gadget writer should create variables in the language stylesheet files that reference the correct localized images. As an example of this, a variable to a “stop” icon could be created:

```
<xsl:variable name="com.novell.nps.GadgetX.images.StopIcon">
    <path to gadget's images>/stop_icon_fr_FR.jpg
</xsl:variable>
```

In the gadget stylesheet, this can be referenced using an XSL Attribute Value Template:

```

```

Locating Localized Files

When a user first accesses Novell Portal Services, the portal will determine what language to use based on the browser's language. When the user authenticates to the Portal, the language and locale information will come through a prioritized list that is stored in the user's object in the directory.

Using the user's language information and a routine in the portal, gadget writers will be able to ask for a localized version of a resource file. In addition, the portal will provide the necessary mechanisms to dynamically build the correct set of layout and language stylesheet files to provide the user with the correct language on their device.

The search routine is built upon an interface that allows different search strategies to be selected by the administrator. By default, the portal will search the directory structure for localized files in the following order (default directories have been underlined):

1. <gadget>/skins/<skin name>/devices/<device>/
2. <gadget>/skins/<skin name>/devices/default/
3. <gadget>/skins/default/devices/<device>/
4. <gadget>/skins/default/devices/default/

If a localized file is not found during this process, the portal will perform a second search for the original non-localized file using the same search order. There should never be a case where no file is found. For a gadget to function, the non-localized file *must* be present.

Gadget Configuration Files

The portal requires gadget require to supply an XML file that describes their available settings. This information is used during the installation, configuration, and administration of the gadget. This file is named *AvailableSettings.xml* and will be stored under the portal's WEB-INF directory. This file uses DTDs to provide the strings used to describe the various settings.

The portal supports localized versions of these DTD files as well. DTD files should be localized with the same naming method used throughout this document. **Fig. 2** shows an example of a localized gadget.

```
WEB-INF
├── gadgets
│   └── com.acme.gadget.GadgetX
│       ├── AvailableSettings.xml
│       ├── AvailableSettings.dtd
│       ├── AvailableSettings_en_US.dtd
│       ├── AvailableSettings_pt_BR.dtd
│       ├── AvailableSettings_fr_FR.dtd
│       ├── AvailableSettings_it_IT.dtd
│       ├── AvailableSettings_de_DE.dtd
│       └── AvailableSettings_es_ES.dtd
```

Figure 2. This shows an example of a typical gadget, whose *AvailableSettings* has been localized. These files are placed under the WEB-INF directory to prevent them from automatically being published by the web server.

The administration components of the portal that need access to these files will use the same routine provided by the portal to acquire localized versions of these files.

Visual SourceSafe Explorer - SourceSafe

File Edit View SourceSafe Tools Web Help

All projects: \$/

- NPS1.5
 - bhkit
 - docs
 - Gadget Config Settings
 - gadgets
 - Install
 - JSSE
 - lib
 - MMM
 - nps
 - admin
 - packages
 - portal
 - support
 - WEB-INF
 - classes
 - lib
 - src
 - com
 - novell
 - nps
 - authentication
 - configManager
 - debug
 - event
 - file
 - gadgetManager
 - gadgets
 - i18n
 - logging
 - pi
 - rendering
 - rmi
 - runtime

\$/.../nps/WEB-INF/src/com/novell/nps/i18n

Name	User	Date-Time	Check Out F
ResourceFileMap.java		2/05/02 10:26a	
SkinPrioritySearchStrategy.java		1/26/02 12:58a	
ResourceFileFilter.java		1/22/02 4:57p	
GadgetResourceFileMap.java		12/20/01 5:17p	
XiffUtil.java		11/21/01 11:21a	
ResourceFilePath.java		6/20/01 3:33p	
ToXiff.java		6/05/01 10:18a	
SystemResourceFileMap.java		5/25/01 4:25p	
ResourceFileSearchStrategy.java		5/21/01 10:18a	

History of \$/.../nps/WEB-INF/src/com/novell/nps/i18n/SystemResourceFileMap.java

History: 15 items

Version	User	Date	Action
	Jbell	2/06/02 1:48p	Labeled 'NPS1.5.2002.2.6'
	Jbell	1/30/02 6:39p	Labeled 'NPS1.5.2002.1.30'
	Jbell	1/29/02 6:40p	Labeled 'NPS1.5.2002.1.29'
	Jbell	1/25/02 5:41p	Labeled 'NPS1.5.2002.1.25'
	Jbell	1/23/02 5:19p	Labeled 'NPS1.5.2002.1.23a'
	Jbell	1/23/02 12:46p	Labeled 'NPS1.5.2002.1.23'
	Jbell	1/22/02 6:33p	Labeled 'NPS1.5.2002.1.22'
	Jbell	1/18/02 4:49p	Labeled 'NPS1.5.2002.1.18'
	Jbell	1/17/02 9:51p	Labeled 'NPS1.5.2002.1.17'
	Jbell	1/15/02 7:40p	Labeled 'NPS1.5.2002.1.15a'
	Jbell	12/10/01 11:53a	Labeled 'NPS1.5C10'
	Jbell	11/12/01 3:10p	Labeled 'NPS1.5b12'
	Jbell	8/20/01 6:17p	Labeled 'NPS1.5-820'
2	Jbell	7/10/01 2:23p	Branched at version 2
1	Btimothy	5/29/01 2:45p	Created

Ready: |norman | Sort: Date | 9 Items

Start | WPWIN | nps on 'jim2.provo.novell...' | Visual SourceSafe Expl... | 10:32 AM

EXHIBIT C

Page 1 of 1

http://www.novell.com/news/press/pressroom/news_brief/archive/2001/07/pr24.html

Novell.

News Brief

Novell Portal Services 1.0 Service Pack 1 Now Available

PROVO, UT — July 24, 2001 — Service Pack 1 for Novell Portal Services 1.0 - eBusiness software that personalizes, secures and focuses a user's business and key relationships - is now available. Novell Portal Services removes the barriers between intranets, extranets, the Internet, and wired and wireless networks while presenting users a single, unified view to the Net.

Novell Portal Services boosts employee productivity and enhances relationships with customers and partners by providing personalized, single-view access to files, collaboration tools, information and applications from all over the Net. With Novell Portal Services, users login with one-step authentication to gain browser-based access anytime, anywhere to applications and information based on their business context (roles, identity, location, workgroups, business hierarchy and any other identifying information that exists on the Net). Ultimately, Novell Portal Services raises productivity for a mobile workforce.

Features Service Pack 1 adds to Novell Portal Services 1.0 include:

- support for internationalization
- ease-of-use enhancements to administrative functions
- three- to four-fold performance increases in scalability, speed and user capacity

Novell Portal Services is built on industry standards like Java, XML, HTML and LDAP, and it runs on eBusiness platforms such as Linux, NetWare, Solaris and Windows NT/2000. Novell Portal Services 1.0 Service Pack 1 is available for free download at <http://www.novell.com/download>.

Press Contacts:

Kevan Barney

Novell, Inc.

Phone: 801-861-2931

E-mail: kbarney@novell.com

[Back](#) ↩

http://www.novell.com/news/press/pressroom/news_brief/archive/2001/07/pr24.html